

REMARKS

## I. INTRODUCTION

In response to the Office Action dated April 21, 2004, the claims have not been amended. Claims 1-36 remain in the application. Re-consideration of the application is requested.

## III. NON ART REJECTION

On page (2) of the Office Action, claims 1-24 were rejected under 35 U.S.C. §112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicants regard as the invention. Specifically, the Office Action provides that claims 1 and 13 fail to provide the link/mapping of the claimed a first, a second portable stylization processes and the claimed a second, a third computer systems and therefore the claims are indefinite.

MPEP 2172.01 provides:

In addition, a claim which fails to interrelate essential elements of the invention as defined by applicant(s) in the specification may be rejected under 35 U.S.C. 112, second paragraph, for failure to point out and distinctly claim the invention. See *In re Venezia*, 530 F.2d 956, 189 USPQ 149 (CCPA 1976); *In re Collier*, 397 F.2d 1003, 158 USPQ 266 (CCPA 1968). But see *Ex parte Nolden*, 149 USPQ 378, 380 (Bd. Pat. App. 1965) ("[I]t is not essential to a patentable combination that there be interdependency between the elements of the claimed device or that all the elements operate concurrently toward the desired result"); *Ex parte Huber*, 148 USPQ 447, 448-49 (Bd. Pat. App. 1965) (A claim does not necessarily fail to comply with 35 U.S.C. 112, second paragraph where the various elements do not function simultaneously, are not directly functionally related, do not directly intercooperate, and/or serve independent purposes.).

Applicants assume that this section sets forth the grounds for the rejection provided in the Office Action. Applicants note that the MPEP provides that the elements do NOT have to be directly functionally related, nor do they have to directly intercooperate.

In view of the above, Applicants traverse the rejection under 35 USC §112, second paragraph.

With respect to claim 1, the first portable stylization process stylizes data of a domain object into an application object. The second portable stylization process stylizes the data of the application object (from the first portable stylization process) into a presentation object. Accordingly, the application object from the first portable stylization step is used in the second portable stylization process. Such a common application object clearly provides a sufficient link/mapping between the claim elements.

Similar to claim 1, claim 13 provides for storing an application object in the memory of a second computer system. The application object is then stylized into a presentation object which is stored in the memory of a third computer system. Thus, the different computer systems are linked by the stylization between the various objects and the storage of the results in respective memories of the computer systems.

Thus, Applicants respectfully request withdrawal of the rejection under 35 USC §112.

#### IV. PRIOR ART REJECTIONS

On page (3) of the Office Action, claims 1-36 were rejected under 35 U.S.C. §102(e) as being anticipated by Helgeson et al., U.S. Patent No. 6,643,652 (Helgeson).

Applicants respectfully traverse these rejections.

Specifically, independent claim 13 was rejected as follows:

As to claim 13, Helgeson et al. (hereinafter referred as Helgeson) discloses a system as claimed by applicant for stylizing data (or transform data) [e.g., see Abstract, lines 3-9; col. 2, lines 51-67] in a computer network system [e.g. see, Fig. 1], comprising:

a) an objected-oriented computer system having memory and a data storage device coupled thereto [e.g., see 211, 209, 217, 219, 221, 223, Fig. 2; col. 5, lines 13-41];

b) a domain object stored in the memory of a first computer system, the domain object comprising an object representation of data stored in a database for a domain entity [e.g. the business object of the fgt-dd-class which stored in the meta-data store as a database table (as shown at col. 14, 2a. The Meta-data Store section) of a business Development Kit (BDK) application server computer system (e.g. col. 13, lines 10-18)];

c) a first portable stylization process [the Platform 501 processing, Fig. 5 and associated texts starting at col. 6, line 26 at seq.] configured to stylize the domain object into an application object [e.g., the BDK (519), Fig. 5 and associated text starting at col. 6, line, 32 at seq.] wherein the application object is stored in the memory of a second computer system [e.g., the platform 501, Fig. 5], the application object comprising an object representation of the data in the domain object that is relevant for a particular computer application [e.g., col. 6, lines 32-60];

d) a second portable stylization process [e.g., WDK server processing (523), Fig. 5 and associated texts starting at col. 6, line 43 at seq.] configured to stylize the application object into a presentation object [e.g., HTML or WML object, col. 6, line 48], wherein the presentation object is stored in the memory of a third computer system [e.g., the client computer system, 515, Fig. 5], the presentation object comprising an object [e.g. the XML, XSL object] representation that encapsulates a visual appearance of the data in the application object [e.g., Fig. 5 and associated texts starting at col. 5, line 54 at seq.].

Applicants traverse the above rejections for one or more of the following reasons:

- (1) Helgeson fails to teach, disclose or suggest three discrete objects – a domain object, an application object, and a presentation object that are based on each other as claimed;
- (2) Helgeson fails to teach, disclose or suggest a portable stylization process that is configured to stylize the domain object into an application object;

- (3) Helgeson fails to teach, disclose or suggest an application object that is an object representation of data in a domain object that is relevant for a particular computer application;
- (4) Helgeson fails to teach, disclose or suggest a portable stylization process configured to stylize an application object into a presentation object; and
- (5) Helgeson fails to teach, disclose or suggest a presentation object that encapsulates a visual appearance of data in an application object.

Independent claims 1, 13, and 25 are generally directed to the ability stylize data in discrete portable steps. As used in the claims and set forth in the specification, "stylization" refers to the process wherein data is transformed from its pure/raw form to the final presentation desired by an application (see page 3, line 22- page 4, line 2). The portability of the stylization process (as claimed) allows stylization to be spread across multiple computers or tiers in a client-server environment.

The claims initially provide for obtaining a domain object. The domain object provides an object representation of data stored in a database for a domain entity.

A portable stylization process then provides the ability to stylize the data in the domain object into an application object. The application object is an object representation of the data in the domain object that is relevant for a particular computer application.

Another portable stylization process then stylizes the application object into a presentation object. The presentation object is an object representation that encapsulates a visual appearance of the data in the application object.

Thus, with the two portable stylization processes and the three different objects (i.e., the domain object, application object, and presentation object), stylization is broken up into multiple discrete parts that provides considerable flexibility.

The cited reference does not teach nor suggest these various elements of Applicants' independent claims. Nor does the cited reference provide the flexibility offered by the present invention as claimed.

Helgeson merely describes a mechanism for managing data exchange among systems in a network. The systems and methods of the present mechanism translate data from a system specific local format to a generic interchange format object, and vice versa, with predefined stylesheets using generic components and a system specific service components which utilize a native application programming interface of the specific local system (see Abstract).

The Office Action relies on the Applications 507 depicted in FIG. 5 to teach the application object as claimed. However, as noted above, the claimed application object is an object representation of data in the domain object that is relevant for a particular computer application. In addition, the application object is created or stylized by a portable stylization process. However, as indicated in col. 6, lines 4-10, application layer 507 merely "provides objects and services particular to a given application." There is no provision in Helgeson that indicates that there are three separable objects as claimed (i.e., a domain object, application object, and presentation object). Further, Helgeson fails to indicate that a portable stylization process created the application object (as claimed). Instead, Helgeson merely indicates that an application layer provides application specific objects and services.

In response to the above arguments, the Office Action now refers to platform 501 to teach the process and Fig. 5, associated text starting at col. 6, line 26. Further, the Office Action equates the application object to the BDK 519 of Fig. 5 and the associated text starting at col. 6, line 32.

The BDK is described in col. 6, lines 32-42 and provides:

BDK (Business Development Kit) Business applications server 519 is Saba's EJB compatibility layer. It extends the standard Java business component model with SABA-specific enhancements, such as improved security and caching, as well as providing an abstraction layer to improve portability between EJB servers. The BDK 519 defines the following base interfaces:  
ISabaEntityBean--The abstraction of a persistent object  
ISabaSessionBean--The abstraction of a transactional service

Thus, as described, the BDK is a kit that merely provided a set of interfaces. The BDK merely extends the standard Java business component model with enhancements for Hegelson's system platform architecture (SABA). However, there is no description, implicit or explicit that the BDK is an application object that comprises an object representation of the data in the domain object that is relevant for a particular computer application. Further, there is no description or suggestion, implicit or explicit, that the application object is produced as a result of a portable stylization process from the domain object.

In rejecting the claims, the Office Action provides that the domain object is equivalent to a business object that is stored in a meta-data store section (as a table) of the BDK. The Office Action then provides that platform model 501 (which contains the BDK 519 (see FIG. 5)) stylizes the object from the database into an application object. However, nowhere is there any description within the cited portion (or the remainder of Helgeson) that provides that the BDK creates an object specific for a particular computer application. Further, Helgeson completely fails to teach a

portable process that stylizes the domain object (or using Hegelson's terms – a business object) into an application object. Instead, the BDK merely describes a kit with various defined interfaces. Such interfaces are not equivalent, implicitly or explicitly to the claimed domain object, stylization process, or application object.

The Office Action continues and rejects the second portable stylization process on the WDK server 523. The Office Action provides that the WDK builds a presentation object in the form of HTML or a WML object. The Office Action then changes its analysis and determines that the presentation object is an XML or XSL object representation. However, in analyzing the claims, one cannot merely ignore the various limitations. In the claims, the application object that is stylized from the domain object is then stylized into the presentation object. Further, the stylization into the presentation object is performed by a second portable stylization process. In Hegelson, the WDK is still part of the same BDK. Further, the WDK states that it generates web content into a variety of formats such as HTML and WML using web standards such as XML and XSL. However, there is no description of where the content is being generated from. Again, the claims provide that the presentation object is stylized from the application object that in turn is stylized from the domain object. In Hegelson, the description merely provides for generating web content. There is no description that the Hegelson's web content is generated from an application specific object that in turn is generated from a domain object. In this regard, when analyzing the claim, the entire sequence (i.e., each and every element) of the claim and the links between the claims must be analyzed when attempting to apply Hegelson. Applicants submit that Hegelson fails to the above claim limitations when examined as a whole in accordance with MPEP 2141.01 and MPEP 2141.02.

The Office Action continues and provides that the presentation object is equivalent to an XSL object relying on col. 5, lines 54 et seq. However, contrary to that asserted by the Examiner, XSL is not an object but instead stands for extensible stylesheet language. In fact, an electronic search of Hegelson for the term "XSL object" provides no results whatsoever. Without even mentioning the terms XSL object, Hegelson cannot possibly teach an XSL object as asserted. Applicants note that the prior Office Action relied on XSLT stylesheets in col. 51, lines 31-34 to teach the presentation object. However, similar to an extensible stylesheet language and contrary to that stated in the prior Office Action, an XSLT stylesheet is not an object. Instead, an XSLT

stylesheet is an extensible stylesheet language transformation (see col. 50, lines 24-25). As defined at <http://www.techweb.com/encyclopedia/defineterm?term=xslt> :

XSLT is the processing component of the Extensible Stylesheet Language (XSL). XSLT is widely used to convert XML to HTML for screen display, but can be used to convert to PDF, another XML document or any other format. The conversion is accomplished with an XSLT processor, which transforms the input based on XSLT extensions of the XSL style sheet. XSL statements are also followed. The processor requires an XML parser to separate the XML elements into a tree structure which the processor manipulates. Path is a component of XSL that is used for identifying input, calculating numbers and manipulating characters. See XSL, DOM and SAX.

Such language clearly indicates that a stylesheet is not an object as claimed. Further, the output produced is an XML document or a document in another format and is not an object representation that encapsulates a visual appearance of the data in the application object. In this regard, not only is the XSLT not an object, but the XSLT is not based on an application object as claimed. Further, Helgeson also fails to indicate or use a portable stylization process that stylizes the application object into the presentation object.

In view of the above, Applicants assert that instead of producing a presentation object that encapsulates a visual appearance of data in the application object, Helgeson merely produces an XML file in accordance with an XSL stylesheet. Such a teaching does not teach, disclose, or suggest the creation or stylization of an object as claimed. Further, Helgeson also fails to teach that the resulting XML file is an visual appearance of data in an application object.

In addition, the Office Action appears to state that the presentation object is equivalent to a WML object. However, similar to an XSL stylesheet, WML merely stands for wireless markup language and does not render, refer to, or make up an object in any way, shape or form. Also similar to the term "XSL object", an electronic search of Helgeson for the term "WML object" provides no results whatsoever. Without even mentioning such words, Helgeson cannot possibly teach such an object.

Moreover, the various elements of Applicants' claimed invention together provide operational advantages over Helgeson. In addition, Applicants' invention solves problems not recognized by Helgeson.

Dependent Claims

Applicants also note that in response to arguments with respect to the dependent claims asserted in Applicants' prior response, the Examiner merely ignored the response and reasserted the identical rejection. Accordingly, Applicants reassert such rejections again as follows.

Applicants note that the dependent claims provide further limitations not addressed or suggested by Helgeson. For example, claims 22-24 provide specify the types of domains in which the application may be used. Specifically, the claims provide that the domains may be a mechanical domain, an architecture, engineering, and construction (AEC) domain, or a geographic information system. These limitations further enhance the independent claims and thereby indicate that the data in the domain objects are relevant in those specific domains.

However, claims 22-24 were merely rejected by stating that these domains are a default nature of a domain object in an internet data exchange computer system. Applicants respectfully disagree with and traverse such a statement. Nowhere in Helgeson is there any reference to either a mechanical, AEC, or geographic information environments, implicitly or explicitly. In fact, separate electronic searches of Helgeson for the terms mechanical, AEC, and geographic provide no results whatsoever. Without even mentioning these terms, Helgeson cannot possibly teach these claims. Further, no art was cited that suggests an internet data exchange computer system by default nature provides for such domain fields.

Further, Applicants traverse the statement that the default nature of a domain object in an internet data exchange computer system includes the mechanical, AEC, and GIS entities. There is no foundation or support for such a conclusory statement. Also, the Office Action fails to cite any art for such a proposition.

Thus, Applicants submit that independent claims 1, 13, and 25 are allowable over Helgeson. Further, dependent claims 2-12, 14-24, and 26-36 are submitted to be allowable over Helgeson in the same manner, because they are dependent on independent claims 1, 13, and 25, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-12, 14-24, and 26-36 recite additional novel elements not shown by Helgeson.

## V. CONCLUSION

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicants' undersigned attorney.

Respectfully submitted,

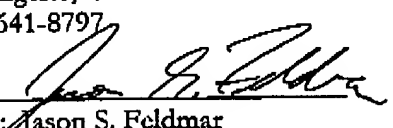
Daniel Lee Thompson et al.

By their attorneys,

GATES & COOPER LLP

Howard Hughes Center  
6701 Center Drive West, Suite 1050  
Los Angeles, California 90045  
(310) 641-8797

Date: June 21, 2004

By:   
Name: Jason S. Feldmar  
Reg. No.: 39,187